Published on *Online Documentation for Altium Products* (https://www.altium.com/documentation)

Using Altium Documentation
Modified by Jason Howie on Apr 11, 2017

Altium Designer's PCB Editor is a 'rules-driven' environment. Design requirements are enforced through a well-defined set of Design Rules, which collectively define the constraints for the board. Design rules target specific objects within a design. For the PCB rules system to know what objects a given rule applies to, it needs to know the scope of that rule - the extent of its application. *Scoping*, or targeting the rule, is performed in the PCB Rules and Constraints Editor dialog.

Scoping a Design Rule

Rather than being restricted to a predefined list of possible target options, each design rule is scoped by writing what is called a *Query*. A query is essentially an instruction to the software that defines the set of design objects to be targeted. Queries are written using query keywords. In the same way that a query can be written in a *Filter* panel to find a specific set of objects, so to a query is written to define the objects that each rule targets. An example might be:

```
InNet('VBAT') And OnLayer('Bottom Layer')
```

If this query were to be used as the scope for a Width rule, whenever you route the VBAT net and switched to the bottom layer, the track width would automatically change to the width specified as part of that rule's constraints. Also, upon running a design rule check, any VBAT net routing on the bottom layer would have to have the specified width, or it would be flagged as a violation.

> Use the options available in the PCB Rules and Constraints Editor dialog to build the query as required. Depending on whether the rule is unary or binary, you will need to define one or two scopes respectively.

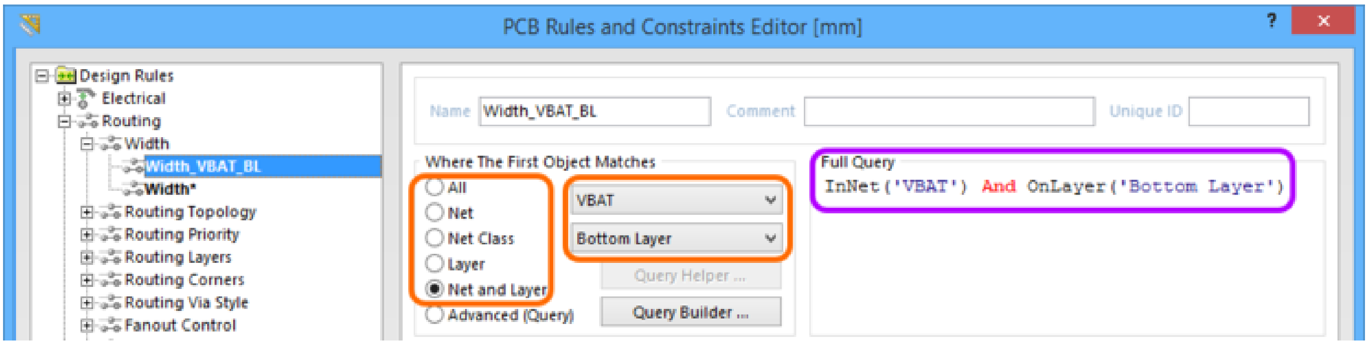
⚠ If there is a syntax error with the query, the rule will be deemed invalid, and will be highlighted in red - within the *PCB Rules and Constraints Editor* dialog - both in the left-hand rule tree, and any summary view (rule category, or rule type) in which the rule appears. A warning message will also appear should the designer attempt to close this dialog. A rule scope that has a syntax error can greatly slow the Online and Batch DRC analysis process - be sure to resolve any rule scopes that are not syntactically correct.

Scoping Options

Simple scoping options are provided that allow you to quickly generate scope queries targeting:

- All design objects.
- All objects in a specific net.
- All objects in a specific net class.
- All objects on a specific layer.
- All objects in a specific net and on a specific layer.

Simply select one of the options and, if required, use the applicable drop-down list(s) to select the appropriate target, such as a Net or Layer. The query string (logical query expression) will appear in the **Full Query** region to the right.



Name Width_VBAT_BL
Comment
Unique ID

Where The First Object Matches
○ All
○ Net
○ Net Class
○ Layer
● Net and Layer
○ Advanced (Query)

VBAT
Bottom Layer

Full Query
```
InNet('VBAT') And OnLayer('Bottom Layer')
```

Quickly create a query using the simple scoping options.

The **Advanced (Query)** option enables you to write your own, maybe more complex, but also more specific query. You can type your own specific query for the rule scope directly into the **Full Query** region. Alternatively, two features are available to help in the creation of logical query expressions - the Query Builder and the Query Helper. These can be very useful when unsure of the syntax of a query, or the possible keywords available for use.

> The *Query Builder* is a simpler method of constructing a query, using sensitive condition types and values that only allow you to build using relevant 'building blocks'. For advanced query construction, with full keyword specification and operator syntax, use the *Query Helper*.

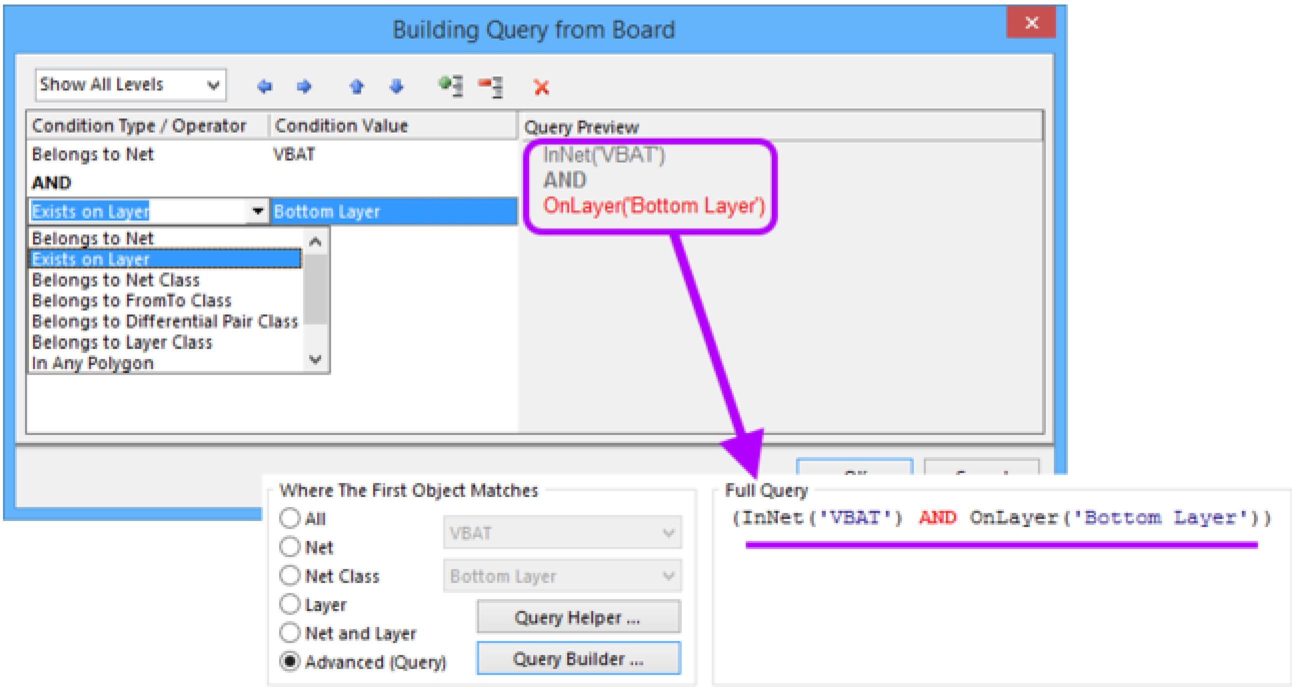
Scoping a Rule with the Query Builder

Click the **Query Builder** button to open the [Building Query from Board dialog](), which enables you to create a query for targeting specific objects in the design document, by simple construction of a string of conditional statements.

The left-hand section of the dialog is used to specify the condition(s) required to target the set of objects needed. Each condition is made up of a *Condition Type*, and a *Condition Value*. Drop-down fields for these populate with entries that are relevant to building a scope for the current rule type, and values for the chosen condition type, respectively.

As a condition is defined, a preview of the currently built query is shown in the right-hand section of the dialog. Conditions can be ANDed and/or ORed together, and also precedence incorporated (through the use of brackets/indenting), to refine the target set of objects. When the expression for the query has been defined as required, clicking **OK** will load the expression into the **Full Query** region of the [PCB Rules and Constraints Editor dialog]().

> Note that additional opening and closing brackets are added to the start and end of the Full Query expression, respectively. Note also that the scoping option will automatically be switched to **Advanced (Query)**.



Condition Type / Operator	Condition Value
Belongs to Net	VBAT
AND	
Exists on Layer	Bottom Layer

Query Preview

InNet('VBAT')
AND
OnLayer('Bottom Layer')

Where The First Object Matches

○ All
○ Net
○ Net Class
○ Layer
○ Net and Layer
● Advanced (Query)

VBAT
Bottom Layer

Full Query

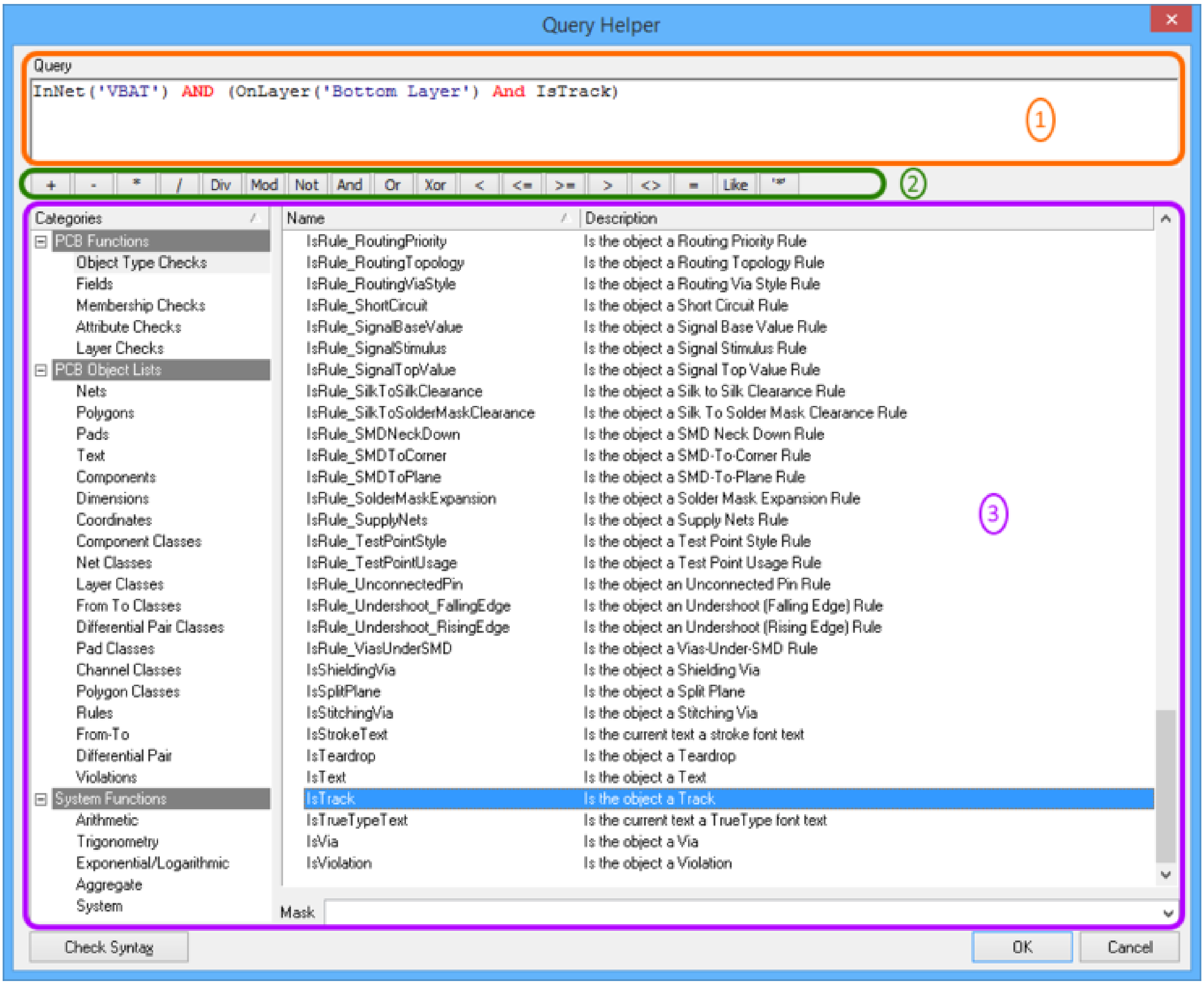
```
(InNet('VBAT') AND OnLayer('Bottom Layer'))
```

Use the *Query Builder* to build a logical query expression through a series of conditional statements.

Scoping a Rule with the Query Helper

With the **Advanced (Query)** option enabled, click the **Query Helper** button to access the [Query Helper dialog](). The underlying Query Engine analyzes the PCB design and lists all available objects, along with generic keywords for use in queries.

Working with the *Query Helper* is fairly intuitive, even more-so when you break it down into three distinct areas, as illustrated in the image below:



```
InNet('VBAT') AND (OnLayer('Bottom Layer') And IsTrack)
```

Categories

- PCB Functions
  - Object Type Checks
  - Fields
  - Membership Checks
  - Attribute Checks
  - Layer Checks
- PCB Object Lists
  - Nets
  - Polygons
  - Pads
  - Text
  - Components
  - Dimensions
  - Coordinates
  - Component Classes
  - Net Classes
  - Layer Classes
  - From To Classes
  - Differential Pair Classes
  - Pad Classes
  - Channel Classes
  - Polygon Classes
  - Rules
  - From-To
  - Differential Pair
  - Violations
- System Functions
  - Arithmetic
  - Trigonometry
  - Exponential/Logarithmic
  - Aggregate
  - System

Name	Description
IsRule_RoutingPriority	Is the object a Routing Priority Rule
IsRule_RoutingTopology	Is the object a Routing Topology Rule
IsRule_RoutingViaStyle	Is the object a Routing Via Style Rule
IsRule_ShortCircuit	Is the object a Short Circuit Rule
IsRule_SignalBaseValue	Is the object a Signal Base Value Rule
IsRule_SignalStimulus	Is the object a Signal Stimulus Rule
IsRule_SignalTopValue	Is the object a Signal Top Value Rule
IsRule_SilkToSilkClearance	Is the object a Silk to Silk Clearance Rule
IsRule_SilkToSolderMaskClearance	Is the object a Silk To Solder Mask Clearance Rule
IsRule_SMDNeckDown	Is the object a SMD Neck Down Rule
IsRule_SMDToCorner	Is the object a SMD-To-Corner Rule
IsRule_SMDToPlane	Is the object a SMD-To-Plane Rule
IsRule_SolderMaskExpansion	Is the object a Solder Mask Expansion Rule
IsRule_SupplyNets	Is the object a Supply Nets Rule
IsRule_TestPointStyle	Is the object a Test Point Style Rule
IsRule_TestPointUsage	Is the object a Test Point Usage Rule
IsRule_UnconnectedPin	Is the object an Unconnected Pin Rule
IsRule_Undershoot_FallingEdge	Is the object an Undershoot (Falling Edge) Rule
IsRule_Undershoot_RisingEdge	Is the object an Undershoot (Rising Edge) Rule
IsRule_ViasUnderSMD	Is the object a Vias-Under-SMD Rule
IsShieldingVia	Is the object a Shielding Via
IsSplitPlane	Is the object a Split Plane
IsStitchingVia	Is the object a Stitching Via
IsStrokeText	Is the current text a stroke font text
IsTeardrop	Is the object a Teardrop
IsText	Is the object a Text
IsTrack	Is the object a Track
IsTrueTypeText	Is the current text a TrueType font text
IsVia	Is the object a Via
IsViolation	Is the object a Violation

Example complex query defined using the *Query Helper*.

These areas are:

1. **Query Region** - use this region of the dialog to compose a query expression. The expression currently in-effect for the rule scope will be made available in this region, by default, when the dialog is accessed. You can type directly within the region; a context-sensitive 'prompt list' of possible keywords or objects will appear as an aid.
2. **Operators** - this region of the dialog provides a range of mathematical and logical operators for use when constructing an expression. Simply click a button to add that operator at the current cursor location within the query expression in the **Query** region above.
3. **Categories, Keywords and Objects** - this region of the dialog provides access to available PCB Functions, PCB Object Lists and System Functions, which can be used to create the query expression. Clicking on a sub-category within each of these three areas, will display a list of corresponding keywords or objects in the region to the right. Simply locate the keyword or object to be used in the query string, then double-click on its entry; the entry will be inserted at the current cursor location within the query expression in the **Query** region above.

Use the **Check Syntax** button (bottom left of the dialog) to verify that an expression is syntactically correct. When the expression for the query has been defined as required, clicking **OK** will load the expression into the **Full Query** region of the PCB Rules and Constraints Editor dialog.

✓ When using the *Query Helper* dialog to construct a query, clicking on a keyword in one of the

✓ available lists and pressing **F1**, will access the documentation for that particular keyword, within the Query Language Reference.

Query Expression Operator Precedence

Before leaving this area, it is worthwhile taking a look at the order of precedence in place for the operators used in logical Query expressions. After all, without such knowledge, an expression may not target the objects required.

Brackets have the highest precedence within an order of precedence that has been defined for the various operators, and which determines how queries are interpreted by the software (whenever the user has not provided brackets). The sequence of this order, from highest to lowest, is as follows:

1. Brackets
2. Not
3. ^, *, /, Div, Mod, And
4. +, -, Or, Xor
5. =, <>, <, >, <=, >=
6. &&, ||

> This order of precedence is similar to that used in Pascal-type languages. Ambiguities are resolved by working from left to right. Brackets are evaluated from inside to outside and equal levels are evaluated left to right.

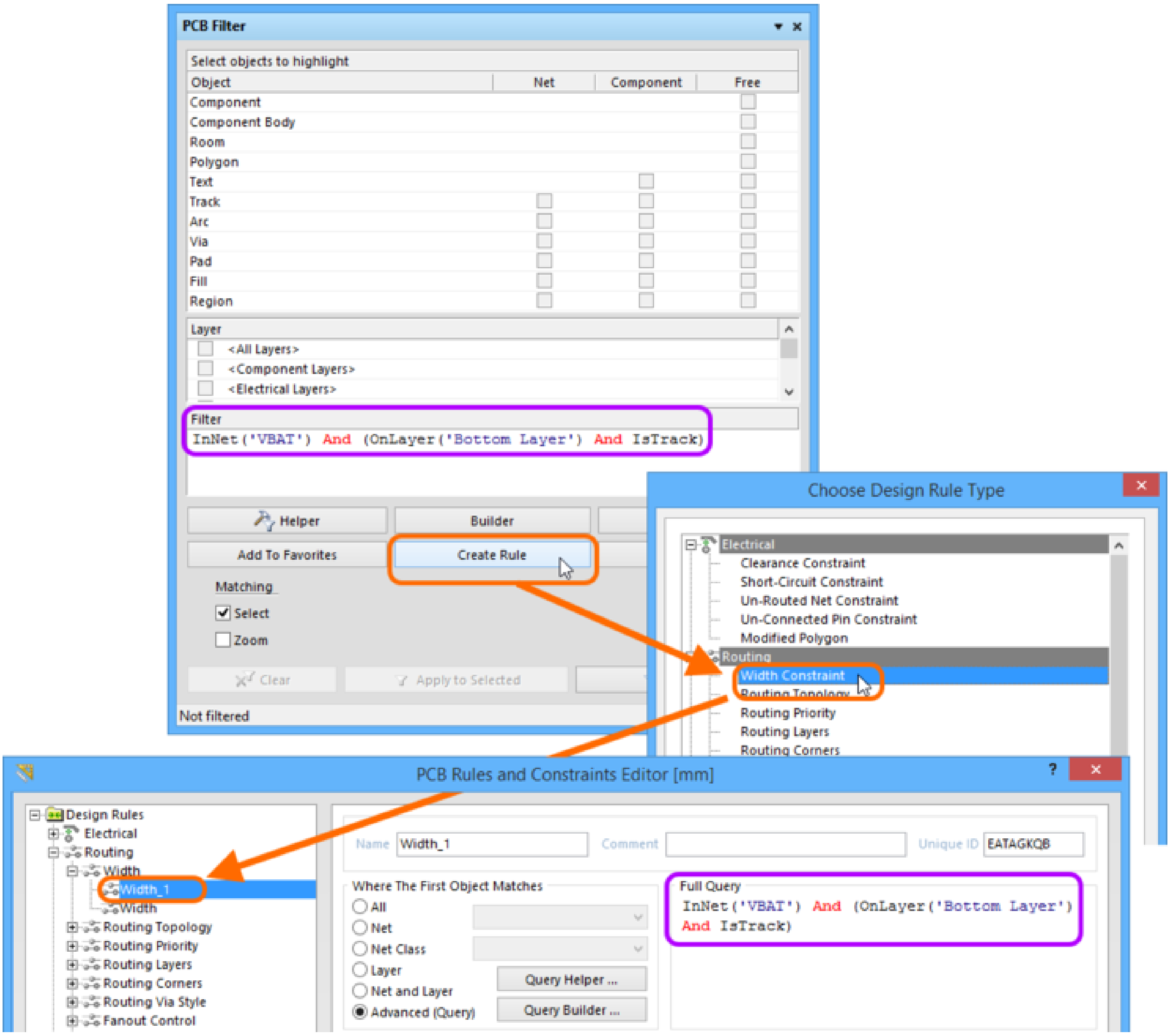
> It is highly advisable to use brackets whenever there is any possibility, whatsoever, that the query might not be correctly interpreted. Generous usage of brackets removes doubt and makes the resulting queries easier to read by others.

Creating a Rule from the PCB Filter Panel

Since a design rule is scoped to apply to a target set of design objects, it can be easier (and faster) to first identify those objects, and then create a design rule that targets them. The PCB Filter panel provides the facility for creating a design rule, the scope of which uses the currently defined query expression in the panel. So rather than struggling to visualize what your query expression *might* target, use the *PCB Filter* panel to test and tweak a query expression until only those required objects are filtered. By then creating a rule that uses that query expression, you are guaranteed to be targeting the correct set of objects.

To add a new design rule:

1. Click on the **Create Rule** button. The Choose Design Rule Type dialog will appear. This dialog lists each of the rule categories and rule types that are available in the PCB document.
2. Choose the type of rule to be created and click **OK** (or double-click directly on the entry).
3. The PCB Rules and Constraints Editor dialog will appear. A rule of the chosen type is created and the main editing window for the rule is displayed, ready to define specific constraints for the rule. The query expression from the *PCB Filter* panel is entered into the **Full Query** region of the dialog.
4. Make changes to the rule's name and constraints as necessary. Also change its priority if needed (it will be given the highest priority by default).



Filter: InNet('VBAT') And (OnLayer('Bottom Layer') And IsTrack)

Matching: ☑ Select □ Zoom

Name: Width_1 Comment: Unique ID: EATAGKQB

Where The First Object Matches: ○ All ○ Net ○ Net Class ○ Layer ○ Net and Layer ● Advanced (Query)

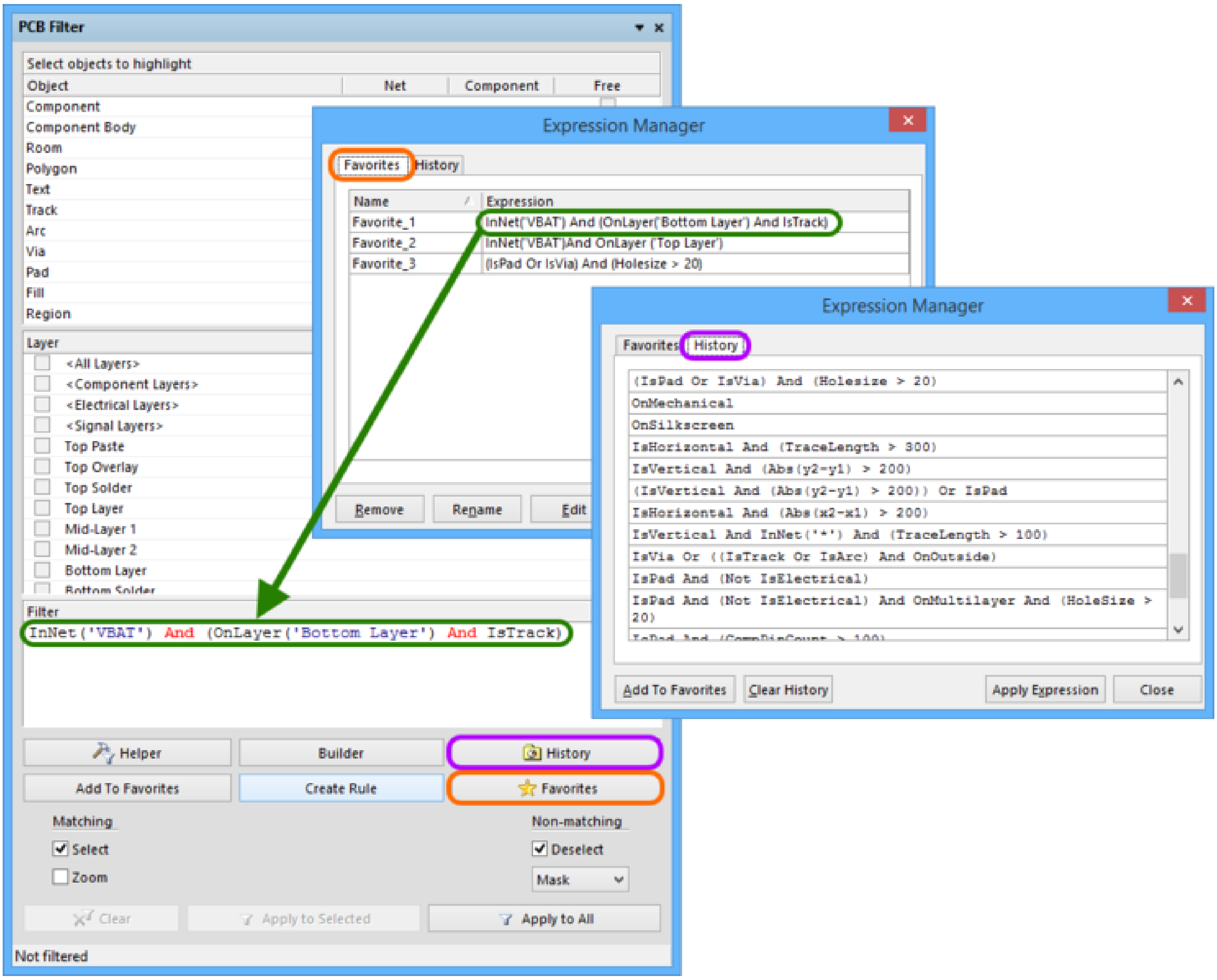
Full Query: InNet('VBAT') And (OnLayer('Bottom Layer') And IsTrack)

Create a new design rule directly from the *PCB Filter* panel - the filter query expression is used as the scope for the rule.

Reusing Past Query Expressions

Over time, a number of useful query expressions will be constructed in the course of laying out various boards. Typically, you will want to apply and re-apply the same queries, not only in the same design, but across different designs. To cater for this, the PCB Filter panel supports the notion of *Historical* and *Favorite* queries.

As a query is entered and applied from the panel, it will be added to a query 'history list'. In addition, that query can be added to a query 'favorites list', by clicking the **Add To Favorites** button. Use the **History** and **Favorites** buttons in the panel to access the corresponding tabs of the same name within the Expression Manager dialog, to see these lists.



Name	Expression
Favorite_1	InNet('VBAT') And (OnLayer('Bottom Layer') And IsTrack)
Favorite_2	InNet('VBAT')And OnLayer ('Top Layer')
Favorite_3	(IsPad Or IsVia) And (Holesize > 20)

```
(IsPad Or IsVia) And (Holesize > 20)
OnMechanical
OnSilkscreen
IsHorizontal And (TraceLength > 300)
IsVertical And (Abs(y2-y1) > 200)
(IsVertical And (Abs(y2-y1) > 200)) Or IsPad
IsHorizontal And (Abs(x2-x1) > 200)
IsVertical And InNet('*') And (TraceLength > 100)
IsVia Or ((IsTrack Or IsArc) And OnOutside)
IsPad And (Not IsElectrical)
IsPad And (Not IsElectrical) And OnMultilayer And (HoleSize > 20)
IsPad And (CompPinCount > 100)
```

```
InNet('VBAT') And (OnLayer('Bottom Layer') And IsTrack)
```

Access a history of applied query expressions and build a listing of favorite expressions to be reused time and again.

To use an expression in either list, simply double-click on its entry, or select its entry and click the **Apply Expression** button. The *Expression Manager* dialog will close and the expression will be loaded into the **Filter** region of the *PCB Filter* panel.

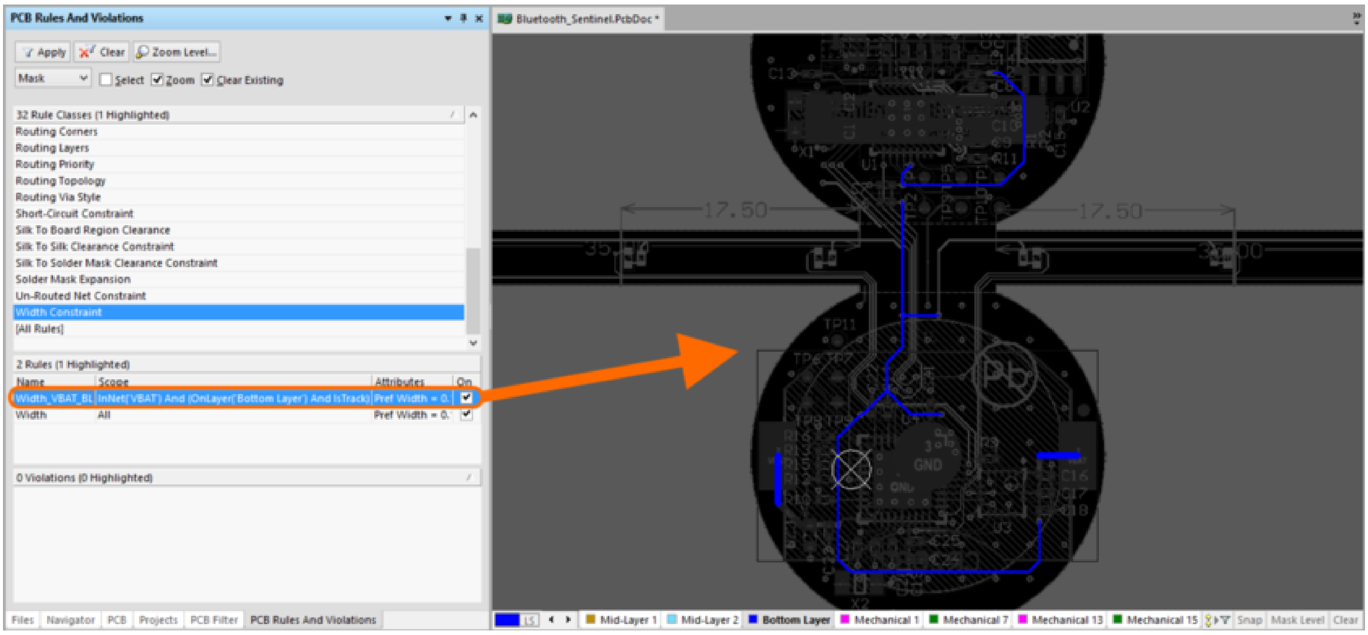
✓ Up to 9 most recently used query expressions from the History list are available from the panel's right-click **History** sub-menu. Up to 10 most recently added query expressions to the Favorites list are available from the top of the right-click menu.

This functionality streamlines (and expedites) creation of rules with requisite scoping - simply retrieve an historical or favorite query expression, check that it still targets the required object set (apply the filter), then create a rule that utilizes that expression in its scoping, as detailed in the previous section.

Checking Rule Application

Depending on the board design, a fair number of design rules may need to be defined, with scopes that range from the very simple, to the very complex. It is a good idea to check that the rules defined do indeed target their intended objects. Care at the rule definition stage can save wasted time and effort tracing violations, caused through incorrect rule scoping.

You can see what objects a particular rule applies to, from the PCB Rules And Violations panel. The panel lists all currently defined rules for the design. All rules can be viewed, or you can browse specifically by rule type - provided at least one rule of any given type has been defined for the active design. As you click on a specific rule in the **Rules** region of the panel, filtering will be applied, using the rule as the scope of the filter. Only those design objects that fall under the scope of the rule will be filtered. By employing the **Mask** (or **Dim**) highlighting feature, you can quickly see the resulting objects targeted by the rule.



Mask ☐ Select ☑ Zoom ☑ Clear Existing

32 Rule Classes (1 Highlighted)

Routing Corners
Routing Layers
Routing Priority
Routing Topology
Routing Via Style
Short-Circuit Constraint
Silk To Board Region Clearance
Silk To Silk Clearance Constraint
Silk To Solder Mask Clearance Constraint
Solder Mask Expansion
Un-Routed Net Constraint
Width Constraint
[All Rules]

2 Rules (1 Highlighted)

Name	Scope	Attributes	On
Width_VBAT_BL	InNet('VBAT') And (OnLayer('Bottom Layer') And IsTrack)	Pref Width = 0.	☑
Width	All	Pref Width = 0.	☑

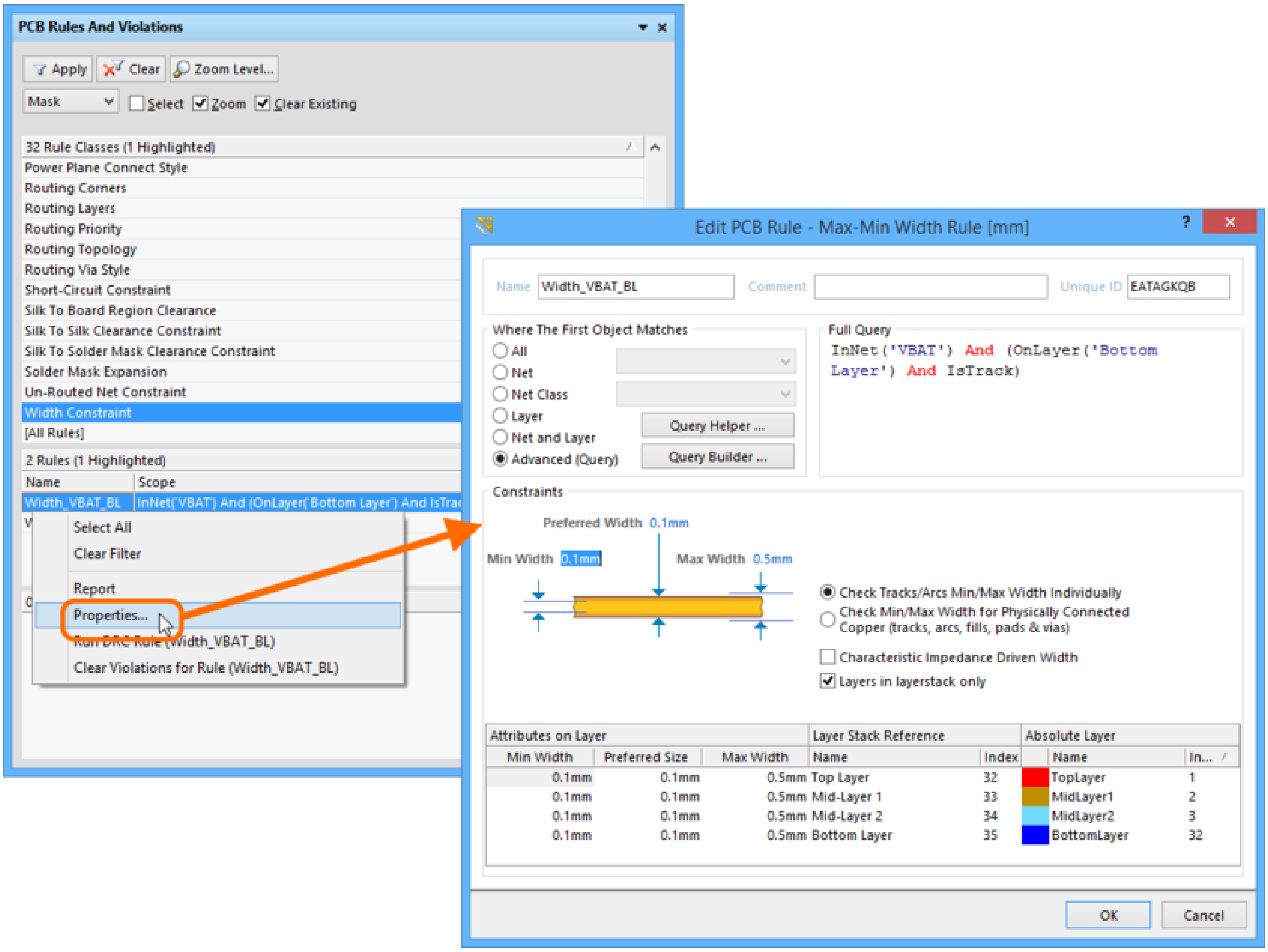
0 Violations (0 Highlighted)

An example of using the *PCB Rules And Violations* panel to check which objects a selected rule applies to.

If the highlighting method in the dialog has been set to `Mask`, adjust the level of masking applied to objects not falling under the scope of the active filter, by using the **Mask Control - Background Objects Factor** slider bar - accessed from the **Mask Level** button (bottom-right of main design window). If the highlighting method in the dialog has been set to `Dim`, adjust the level of dimming applied to objects not falling under the scope of the active filter, by using the **Dim Control - Masked Objects Factor** slider bar - accessed from the **Mask Level** button (bottom-right of main design window).

Using the *PCB Rules And Violations* panel to interrogate which objects a rule applies to is particularly useful when creating a query for a rule’s scope(s), since a rule can be edited directly from the panel, and therefore the query can be 'tweaked' until the desired objects are captured by the scope(s).

To edit the scope(s) for a rule, as well as its constraints, either double-click on the rule's entry in the panel, or right-click and choose **Properties** from the context menu. The relevant Edit PCB Rule dialog will appear, from where the changes can be made.



PCB Rules And Violations

Apply | Clear | Zoom Level...

Mask | □ Select | ☑ Zoom | ☑ Clear Existing

32 Rule Classes (1 Highlighted)
Power Plane Connect Style
Routing Corners
Routing Layers
Routing Priority
Routing Topology
Routing Via Style
Short-Circuit Constraint
Silk To Board Region Clearance
Silk To Silk Clearance Constraint
Silk To Solder Mask Clearance Constraint
Solder Mask Expansion
Un-Routed Net Constraint
Width Constraint
[All Rules]

2 Rules (1 Highlighted)

Name	Scope
Width_VBAT_BL	InNet('VBAT') And (OnLayer('Bottom Layer') And IsTrac

Select All
Clear Filter
Report
Properties...
Run DRC Rule (Width_VBAT_BL)
Clear Violations for Rule (Width_VBAT_BL)

Edit PCB Rule - Max-Min Width Rule [mm]

Name: Width_VBAT_BL | Comment: | Unique ID: EATAGKQB

Where The First Object Matches
○ All
○ Net
○ Net Class
○ Layer
○ Net and Layer
● Advanced (Query)

Query Helper ... | Query Builder ...

Full Query

```
InNet('VBAT') And (OnLayer('Bottom
Layer') And IsTrack)
```

Constraints

Preferred Width 0.1mm
Min Width 0.1mm
Max Width 0.5mm

● Check Tracks/Arcs Min/Max Width Individually
○ Check Min/Max Width for Physically Connected Copper (tracks, arcs, fills, pads & vias)
□ Characteristic Impedance Driven Width
☑ Layers in layerstack only

Attributes on Layer			Layer Stack Reference		Absolute Layer	
Min Width	Preferred Size	Max Width	Name	Index	Name	In...
0.1mm	0.1mm	0.5mm	Top Layer	32	🟥 TopLayer	1
0.1mm	0.1mm	0.5mm	Mid-Layer 1	33	🟫 MidLayer1	2
0.1mm	0.1mm	0.5mm	Mid-Layer 2	34	🟦 MidLayer2	3
0.1mm	0.1mm	0.5mm	Bottom Layer	35	🟦 BottomLayer	32

OK | Cancel

Rules can be edited directly from the *PCB Rules And Violations* panel, allowing you to further refine rule scoping and/or constraints, on-the-fly.

Source URL: https://www.altium.com/documentation/display/ADES/((Scoping+Design+Rules))_AD